Zeszłoroczny cykl publikacji o APPLE IIe wywołał sporo kontrowersji. Wiele osób zarzucało nam marnowanie miejsca mikroKLANU na opisywanie komputera będącego praktycznie poza granicą możliwości finansowych większości czytelników. Z drugiej strony — do autora publikacji (podaliśmy numer telefonu) dzwoniło wiele osób chcących uzyskać bardziej szczegółowe informacje. Często rozważano możliwość zakupu właśnie tego mikrokomputera.

W zeszłym roku pojawiło się kilka nowych, interesujących propozycji, czujemy się więc zobowiązani przynajmniej w sposób skrótowy je opisać. Dla oszczędności miejsca postaramy się, aby opisy konstrukcji były skondensowane, niemniej pozwalające zdobyć wstępne rozeznanie.

APPLE IIc

Wypełniając ubiegłoroczne przyrzeczenie swego szefa Steva Jobs'a („co rok nowy wyrób"), firma APPLE zaprezentowała na wystawie w San Francisco nowy mikrokomputer APPLE IIc (od ang. compact — zwarty). Rzeczywiście, IIc jest konstrukcją bardzo zwartą, ma bowiem wymiary 292×305×57 mm i waży ok. 3,5 kg.

Konstrukcja

APPLE IIc jest zbudowany na bazie mikroprocesora 65C02 (wersja CMOS), który w stosunku do 6502 ma dodatkowe instrukcje pozwalające tworzyć szybsze i bardziej zwarte oprogramowanie. Doświadczenia w projektowaniu i stosowaniu układów scalonych umożliwiły zmniejszenie ich liczby z 31 w IIe do 21 w IIc. Wśród nich znalazł się sterownik dysków elastycznych, zaprojektowany przez Steva Wozniaka (współzałożyciela firmy APPLE) i stąd nazywany IWM (Integrated Woz Machine).

Podstawowe cechy APPLE IIc:
— pamięć wewnętrzna: 128 KB RAM i 16 KB ROM (zawierająca Applesoft BASIC i Mouse Text)
— wbudowana stacja dyskietek 5,25" o pojemności 143 KB
— klawiatura 63-przyciskowa
— grafika 16-kolorowa o rozdzielczości 280×192 punkty lub 6-kolorowa o rozdzielczości 560×192 punkty

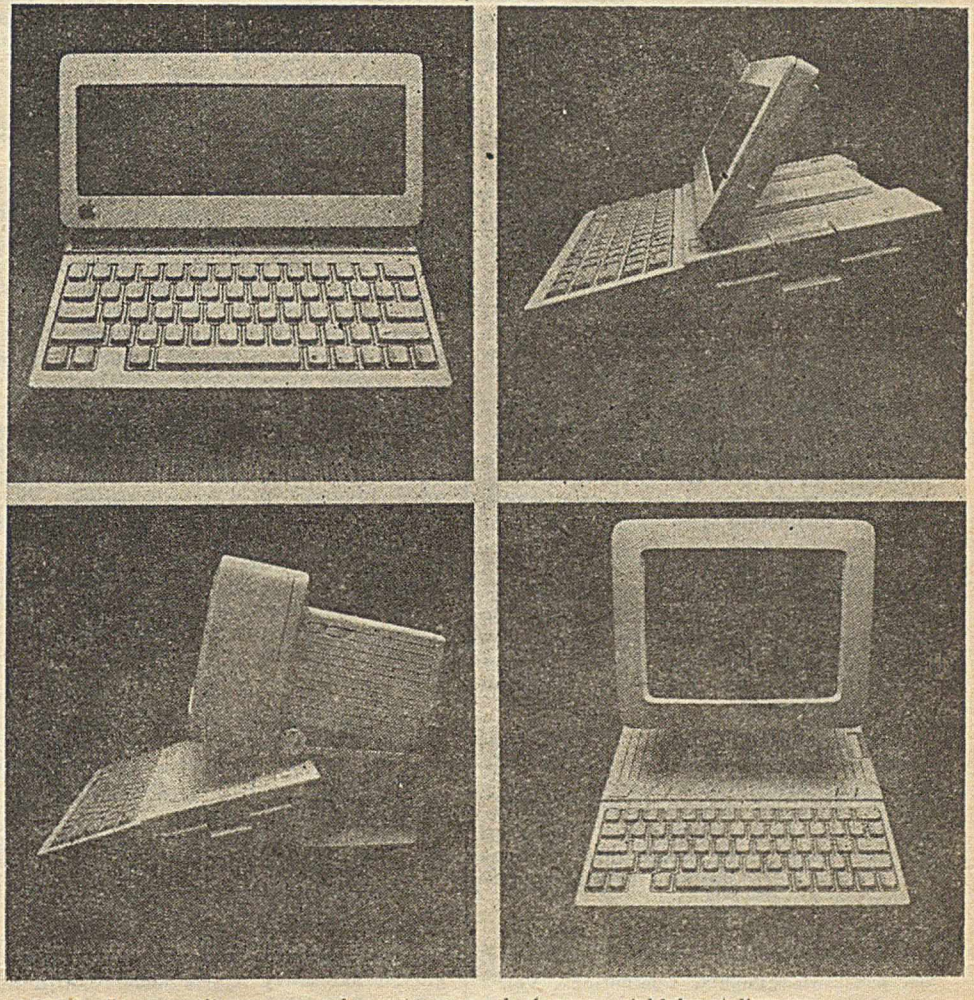
Do umieszczonych z tyłu obudowy gniazd w APPLE IIc można dołączyć myszkę lub manipulator drążkowy, modem (300 lub 1200 bodów), monitor kolorowy lub ekran ciekłokrystaliczny, monitor monochromatyczny, zewnętrzną stację dyskietek i drukarkę szeregową.

Klawiatura APPLE IIc jest bardziej spłaszczona niż w IIe (widać to na zdjęciach). Zastosowano też kilka nowych rozwiązań:
— możliwość przełączenia układu klawiatury ze standardowego QWERTY na układ Dvorak'a, który określa się jako przyszłościowy
— po prawidłowym wciśnięciu klawisza słychać cichy brzęk i daje się odczuć leciutkie drżenie klawisza (tzw. odpowiedź dotykowo-słuchowa)
— pod klawiszami zamontowano elastyczną podkładkę (ang. splash pad), która chroni wnętrze systemu przed zwarciem, jeśli na klawiaturę przypadkowo rozleje się jakiś płyn.

Użytkownik ma możliwość wyboru między 40- a 80-kolumnowym tekstem na ekranie. Wynika to z dwóch względów. Po pierwsze — wiele programów dostępnych na rynku napisanych jest z myślą o wyświetlaniu 40 znaków w linii. Po drugie — tekst 80-kolumnowy jest często niewyraźny na ekranie (konieczny jest monitor wysokiej klasy).

Najciekawszym elementem APPLE IIc jest ekran ciekłokrystaliczny, dostarczany przez japońską firmę SHARP. Pozwala on wyświetlać 24 wiersze 80-kolumnowego tekstu oraz przedstawiać grafikę o dużej rozdzielczości. Ekran ów znacznie podnosi walory APPLE IIc jako mikrokomputera przenośnego. Ponadto APPLE IIc może być zasilany napięciem 7...24 V (niebagatelna cecha na rynkach z dużą ilością samochodów i... różnorodnych akumulatorów).

W odróżnieniu od APPLE IIe, model IIc nie daje możliwości rozbudowy sprzętowej (nie ma dostępu do szyny systemowej), co nie pozwala np. na dołączenie pakietów z procesorem Z80 — nie można więc korzystać z bogatego oprogramowania napisanego dla CP/M. Jest to chyba najpoważniejsza wada nowego APPLE.



APPLE IIc z monitorem monochromatycznym i ekranem ciekłokrystalicznym

Oprogramowanie

APPLE IIc pracuje pod nowym systemem operacyjnym ProDOS (Professional Disk Operating System) kompatybilnym z poprzednikami: DOS3.3 (APPLE II) i SOS (Sophisticated Operating System — APPLE III). Główna różnica polega na włączeniu do ROM programu obsługi myszki (ang. Mouse Text), który generuje piktogramy (ang. icons) podobne do tych w mikrokomputerze MACINTOSH. Stąd ok. 10% programów napisanych dla APPLE IIe nie jest w pełni kompatybilnych z IIc i wymaga niewielkich przeróbek. Wiele z nowych programów używa już myszki w miejsce manipulatora drążkowego. Szerzej o oprogramowaniu APPLE pisano już w poprzednich numerach INFORMATYKI.

Nowy styl projektowania (ang. new design language)

Podobnie jak APPLE IIe, model IIc nie ma klawiszy funkcyjnych, by — jak stwierdzono — maksymalnie uprościć posługiwanie się mikrokomputerem. Temu celowi służy również obrazkowe przedstawienie funkcji każdego portu z tyłu obudowy.

APPLE IIc jest rezultatem szeroko zakrojonych badań rynku i zachowań użytkowników komputerów. Firma APPLE dostarczyła próbne egzemplarze specjalnie dobranym grupom użytkowników i zbierała opinie o pracy nowego mikrokomputera.

Okazało się, że użytkownicy małą wagę przywiązują do oprogramowania, uważając, że dobry sprzęt automatycznie gwarantuje sukces. Ponadto użytkownicy zbyt mało czasu poświęcają na dobre zapoznanie się z możliwościami sprzętu (grube książki-przewodniki!), stąd późniejsze kłopoty i — niejednokrotnie — zniecierpliwienie.

Wyciągając stąd wnioski, APPLE dostarcza 125-stronicowy przewodnik objaśniający sześć programów (umieszczonych na dyskietkach), które w sposób konwersacyjny uczą posługiwania się IIc. Programy owe zapoznają użytkownika zarówno z „zabawowym" wykorzystaniem APPLE IIc w grach komputerowych, jak i w poważniejszych przedsięwzięciach (przetwarzanie tekstów, korzystanie z bazy danych czy z programów kalkulacyjnych). Wprowadzają również elementy języków LOGO i BASIC.

Firma APPLE planuje, że 55% spośród wyprodukowanych modeli IIc znajdzie zastosowanie jako komputery domowe, reszta — w szkołach i w biurach. Nie ulega wątpliwości, że firma wiąże z nowym wyrobem duże nadzieje, przede wszystkim — w walce rynkowej z IBM PCjr.

opracował ADAM STAWOWY

na podstawie „Popular Computing", Vol. 3 No. 8

Publikujemy kolejny tekst, który jest wynikiem prac nad udoskonaleniem najtańszego mikrokomputera. Standardowy 1 KB pamięci to tak niewiele, że praktycznie każdy użytkownik po pewnym czasie kupuje rozszerzenie. Wbudowana wtedy w komputerze pamięć RAM marnuje się niewykorzystana. Podstawą prezentowanego w tekście pomysłu jest odzyskanie firmowego kilobajta RAM przy stosowaniu przystawki 16 KB. Można go wykorzystać do zdefiniowania odmiennego zestawu znaków (jak proponuje autor) lub do umieszczenia procedur w języku maszynowym.

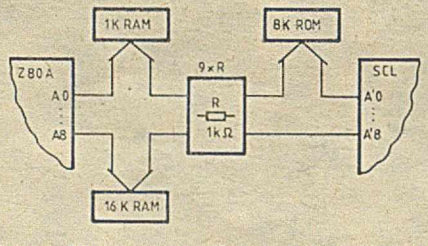
Opisana przeróbka wymaga ingerencji w budowę komputera, powinna więc być wykonywana przez osoby posiadające już pewne doświadczenie. Podjęte ryzyko można zmniejszyć starannie sprawdzając każdy wykonywany krok, natomiast postawa: „żeby tylko nie zepsuć" na pewno nie posunie nas do przodu.

Rozszerzenie możliwości graficznych ZX81 (16K)

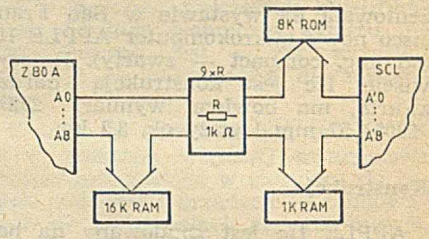
Jednym z głównych ograniczeń komputera ZX81 są jego małe możliwości graficzne. Ponieważ przystawki umożliwiające otrzymanie dużej rozdzielczości (256×192) dotychczas nie uległy w kraju szerszemu upowszechnieniu, proponuję rozwiązanie uproszczone.

Wykonując stosunkowo prostą przeróbkę połączeń wewnętrznych i stosując jeden dodatkowy układ scalony, możemy przesunąć tablicę znaków komputera z ROM-u do RAM-u. Pozwala to na deklarację nowego zestawu znaków, zależnie od potrzeb użytkownika mogą to być np. symbole elektroniczne czy litery innego alfabetu. Ponieważ nie ulega zmianie liczba możliwych znaków, jaką dysponuje komputer (64+64 negatywy), deklarując nowy znak musimy zrezygnować z jednego ze znaków starych — nie otrzymujemy więc możliwości stosowania tak uzyskanej wysokiej rozdzielczości w sposób nieograniczony.

Sposób kreślenia znaków w ZX81 polega na generacji przez SCL (ang. Sinclair Computer Logic) adresów kolejnych wierszy znaku zapisanego w tablicy 8×8 (8 bajtów na każdy znak) [2]. Adres początku tablicy znaków — 7680 pamiętany jest w rejestrze I mikroprocesora Z80. Generowane przez SCL adresy (A'_0—A'_8) nie mogą obciążać wyjść adresowych Z80 i z tego względu konstruktorzy ZX81 zastosowali oporniki maskujące (rys. 1). Oporniki te, dodatkowo, oddzielają wewnętrzną pamięć 1K RAM od SCL. Pamięć ta przy pracy z modułem 16K RAM jest blokowana i aby ją wykorzystać do kreślenia znaków, należy przejąć jej sterowanie (sygnałem RAMCS'/) i połączyć jej wejścia adresowe z adresami A'_0—A'_8 (rys. 2). Omawiane zmiany nie przekreślają możliwości używania komputera w wersji 1K, a dodatkowo umożliwiają wykorzystanie pamięci 1K przy pracy z modułem 16K (otrzymujemy 17K dostępnej pamięci RAM).

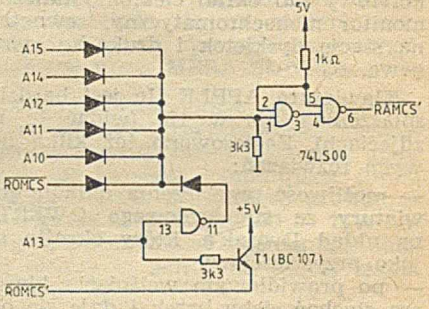


Rys. 1. Uproszczony schemat połączenia magistrali adresowej A_0—A_8 komputera ZX81 16K



Rys. 2. Schemat magistrali adresowej A_0—A_8 komputera po przeprowadzeniu opisywanych zmian

Przełączenie wejść adresowych A_0—A_8 RAM należy wykonać tak, aby nie zmieniły one sygnałów adresowych A_0—A_8 doprowadzonych do modułu 16K. Ze względu na dużą liczbę dodatkowych połączeń, jak też przecięć ścieżek na płytce drukowanej, przed włączeniem komputera należy kilkakrotnie sprawdzić prawidłowość wykonania tych zmian. Potwierdzeniem właściwego ich przeprowadzenia jest dobra praca komputera w wersji 1K. Do modułu 16K należy doprowadzić sygnał ROMCS/, rezygnując z jednego z sygnałów nie używanych przez ten moduł — np. Φ (zegar). Sygnał ten potrzebny jest układowi sterowania pamięci 1K przy pracy w wersji 17K z możliwością definiowania nowych znaków. Schemat układu przedstawia rysunek 3.



Rys. 3. Schemat układu sterowania pamięci 1K RAM

Układ ten należy umieścić wewnątrz modułu 16K RAM, co ze względu na małą liczbę użytych elementów nie powinno sprawić kłopotu. We wtyku modułu należy rozewrzeć połączenie końcówki RAMCS'/ z końcówką +5V i dołączyć do RAMCS'/ wyjście układu sterowania pamięci. Podczas pracy w wersji 17K, pamięć 1K umieszczona jest w obszarze od adresu 8192 do 9215, który został odblokowany przez tranzystor T1 (obszar 8-16K traktowany był przez procesor na równi z zajmowanym przez ROM obszarem 0-8K).

Po sprawdzeniu prawidłowości wykonania układu sterowania (nie należy stosować w nim bramek z serii standardowej), sprawdzamy jego działanie w połączeniu z komputerem (i z modułem 16K), np. za pomocą instrukcji POKE i PEEK. Aby przypisać tablicę znaków komputera obszarowi pamięci

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	36
0	0	1	1	0	1	0	0	52
1	1	1	1	1	1	1	1	255
0	0	1	1	0	1	0	0	52
0	0	1	0	0	1	0	0	36
0	0	0	0	0	0	0	0	0

Rys. 4. Przykład deklaracji nowego znaku

8-9K, należy zmienić, za pomocą krótkiego programu w języku wewnętrznym, wartość pamiętaną w rejestrze I z 30 na 32 lub 34 (nie należy używać 33 lub 35). W pierwszym przypadku początkiem tablicy znaków będzie adres 8192, a w drugim — 8704.

Położenie znaku w tablicy znaków określa jego kod. Znak o kodzie „0" zajmuje pierwsze osiem bajtów, następne osiem bajtów zajmuje znak o kodzie „1", itd. Jako przykład deklaracji nowego znaku, na rysunku 4 przedstawione są wartości kolejnych ośmiu adresów symbolu graficznego diody.

JACEK PSZONA
Warszawa

LITERATURA

[1] Dawidowski J.: Komputer osobisty ZX81. Informatyka, 5/1983
[2] Kaczmarczuk J.: Jeszcze o ZX81. Informatyka, 9/1983

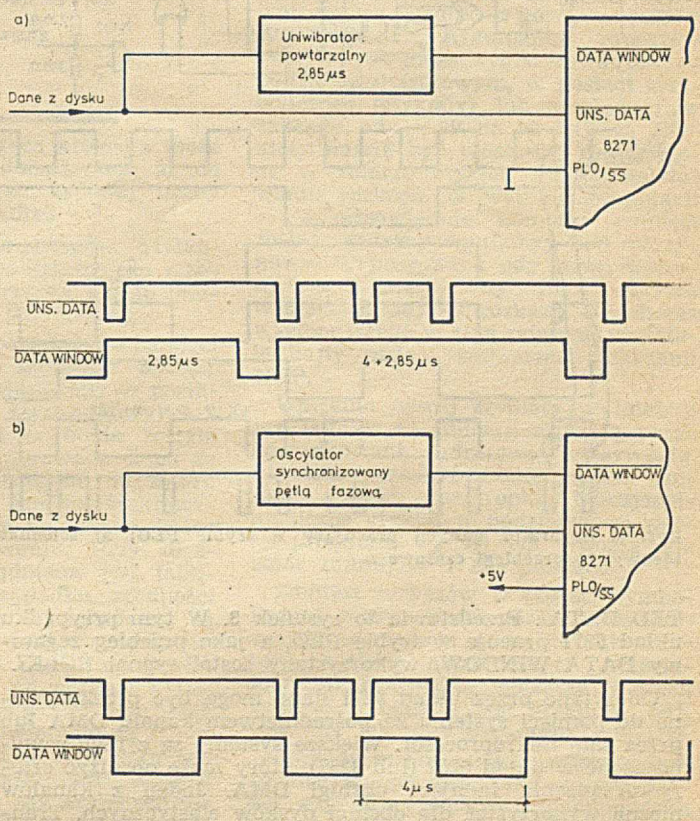
Drugą częścią artykułu o współpracy z polskimi napędami dysków elastycznych kończymy cykl publikacji o tym coraz popularniejszym typie pamięci masowej. Do tematyki dysków elastycznych zapewne jeszcze wrócimy — w chwili, gdy właściciele mikrokomputerów bardziej dojrzeją do tego problemu, również finansowo. Na razie skromne łamy mikroKLANU nie pozwalają na zaspokojenie wszystkich potrzeb. Zachęcamy jednak Czytelników do dzielenia się z nami swoją opinią o przydatności dotychczasowych publikacji. Będzie to bardzo pomocne przy dobieraniu do druku przyszłych materiałów.

Współpraca z napędem dysków elastycznych PL × 45D (2)

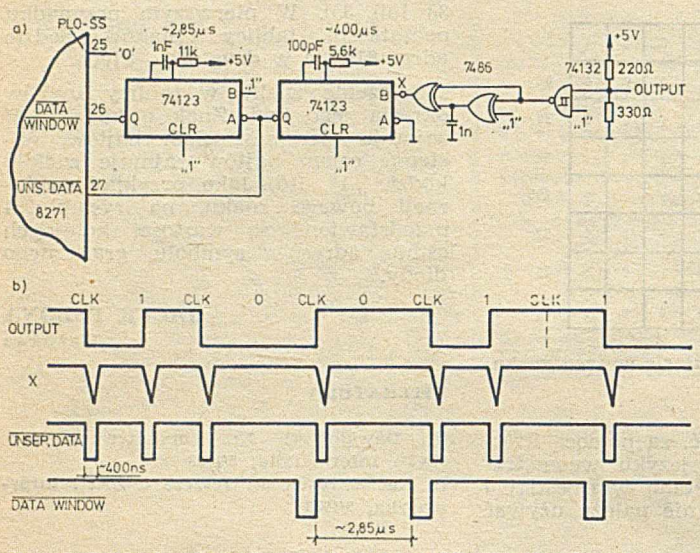
W układzie 8271 przewidziano dwa sposoby dekodowania danych, z sygnału odczytanego z dysku. Do wyboru sposobu służy wejście PLO-SS/. Na wejście UNSEPARATED DATA/ (wprowadzone są impulsy danych i zegara. Ich rozdzielenie możliwe jest dzięki wejściu DATA WINDOW/. Jeżeli na PLO-SS/ podany jest stan logiczny 0, to układ pracuje w trybie SINGLE SHOT. W tym przypadku sygnał DATA WINDOW/ wytwarzany jest z wykorzystaniem powtarzalnego uniwibratora, wyzwalanego od każdego impulsu danych i zegara (patrz rysunek 3 w poprzedniej części artykułu). Uniwibrator generuje impuls, jeżeli przez 2/3 okresu zegara (około 2,85 μs) na linii danych nie pojawi się sygnał. Oznacza to, że brak jest impulsu danych (dane=0) lub brak impulsu zegarowego.

Jeżeli na wejście PLO-SS/ podany jest stan logiczny 1, układ pracuje w trybie PHASE LOCKED OSCILATOR. W tym przypadku sygnał DATA WINDOW/ wytwarzany jest z wykorzystaniem generatora śledzącego częstotliwość sygnału odczytywanego z dysku. Jeżeli poziom wysoki wskazuje na impulsy zegarowe, to impulsy danych pojawiają się przy poziomie niskim. Faza przebiegu zegarowego nie jest istotna. Ważne jest natomiast, by sygnał generatora prawidłowo nadążał za zmianami częstotliwości odczytywanych danych. Oba tryby pracy przedstawiane są na rysunku 1.

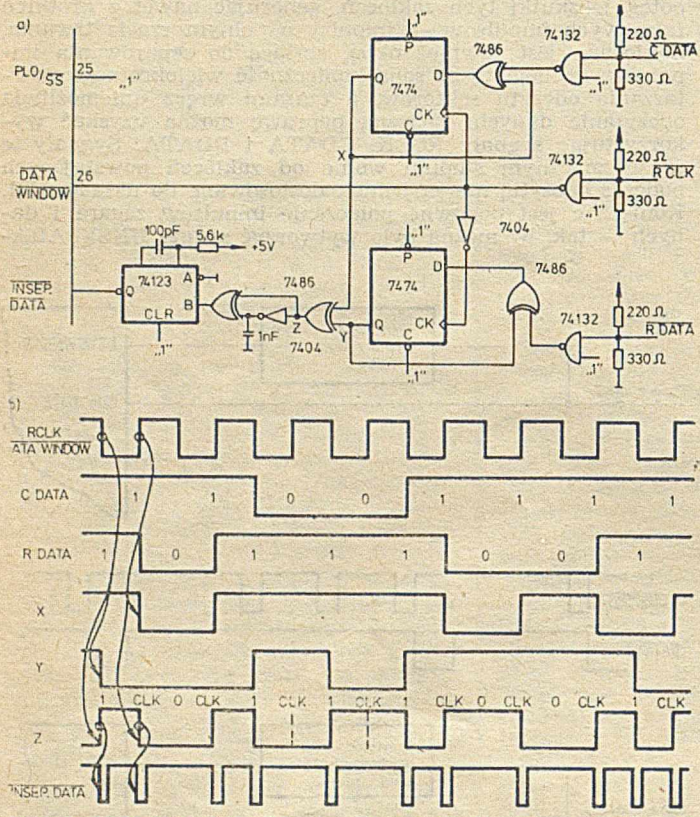
Sygnały odczytu danych z PLx45D można dołączyć do układu 8271 pracującego zarówno w trybie SS/, jak i PLO. W trybie SS/ sygnał UNSEPARATED DATA/ można wytworzyć z sygnału OUTPUT/. W przedstawionym na rysunku 2 układzie każde zbocze sygnału OUTPUT służy do wygenerowania ujemnego impulsu o czasie trwania ok. 400 ns. Impulsy te maskowane są przez kolejny uniwibrator, wytwarzający sygnał DATA WINDOW/. Ten prosty układ nie daje jednak w pełni zadowalających rezultatów. Sygnał OUTPUT/ jest mocno zakłócony, a przedstawiony układ potęguje skutki tych zakłóceń, generując nawet z krótkich fałszywych impulsów — impulsy o pełnym czasie trwania. Zakłócane jest również okno, służące do separowania impulsów. Pociąga to za sobą konieczność wielokrotnego powtarzania odczytu sektorów, a czasami wręcz uniemożliwia odczytanie danych. Znaczną poprawę można uzyskać wykorzystując sygnały RCLK, CDATA i RDATA. Sygnały te są w znacznym stopniu wolne od zakłóceń powstających podczas odczytu, nie są jednak dostosowane do układu 8271. Konieczne jest ponowne połączenie impulsów zegara i danych — tak, by można było wytworzyć sygnał UNSEPARA-



Rys. 1. Sposób separacji danych z sygnału modulowanego FM, a) z wykorzystaniem uniwibratora powtarzającego, b) z wykorzystaniem PLO



Rys. 2. Separator danych pracujący w trybie SS; a) schemat ideowy, b) przebiegi czasowe



Rys. 3. Separator danych pracujący w trybie PLO; a) schemat ideowy, b) przebiegi czasowe

TED DATA/. Przedstawia to rysunek 3. W tym przypadku układ 8271 pracuje w trybie PLO, a jako przebieg zegarowy DATA WINDOW/ wykorzystany został sygnał RCLK/.

Odczytane przez układ 8271 dane mogą być przekazywane do pamięci systemu za pośrednictwem kanału DMA lub przez sam mikroprocesor. Większe systemy są z reguły wyposażone w układ 8257 (lub 8237), który może obsłużyć cztery urządzenia żądające obsługi DMA. Jeden z kanałów można wykorzystać dla obsługi dysków elastycznych. Transmisja, z wykorzystaniem układu DMA znacznie upraszcza program obsługi dysków i sprowadza się do zainicjowania kierunku transmisji oraz adresu początku bufora danych.

Jeżeli w systemie nie ma układu DMA, procesor musi być całkowicie zaangażowany w przesyłanie danych. Czas, w jakim należy odebrać i przesłać dane, jest bardzo krótki, biorąc pod uwagę szybkość działania procesora 8080 oraz liczbę warunków, jaką musi on sprawdzić. Układ 8271 nie może oczekiwać na odebranie lub dostarczenie danych dłużej niż 31 µs. W tym czasie procesor musi sprawdzić waru-

```
                .
                .
                .
                LXI  H,ADDRESS          ; ADRES BUFORA
                MVI  C,RF  OR IRQ       ; MASKA KOŃCA OPERACJI
                MVI  B,CB  OR ND OR IRQ ; MASKA DANYCH
READ:           IN SREG                 ; POBIERZ STATUS   10T
                CMP B                   ; SĄ DANE ?         5T
                JNZ RESULT              ; TESTUJ DALEJ     10T
DATA:           IN DREG                 ; POBIERZ DANE     10T
                MOV M,A                 ; SCHOWAJ DANE      7T
                INX H                   ; NOWY ADRES        5T
                JMP READ                ; KONTYNUUJ        10T
RESULT:         CMP C                   ; KONIEC ?          5T
                JNZ READ                ; KONTYNUUJ        10T
                .
                .
                .
```

(Uwaga: CB=80H, RF=10H, IRQ=8, ND=4)

```
                .
                .
                .
                LXI D,ADDRES            ; ADRES BUFORA
                MVI H,PROGRAM           ; ADRES BAZOWY PROGRAMU
                MVI B,RF OR ND OR IRQ   ; MASKI
                .
                .
                .
READ:           IN  SREG                ; POBIERZ STATUS
                ANA B                   ; WYTNIJ BITY
                MOV L,A                 ; MŁODSZY BAJT BAZY PROGRAMU
                PCHL                    ; SKOK DO PROCEDURY
                .
                .
                .
DATA:           IN  DREG                ; POBIERZ DANĄ
                STAX D                  ; SCHOWAJ DANĄ
                INX D                   ; NOWY ADRES
                JMP READ                ; KONTYNUUJ
                .
                .
                .
RESULT: IN RREG                         ; POBIERZ WYNIK
                .
                .
                .
```

nek gotowości danych, warunek zakończenia operacji, odebrać dane i zapisać je do pamięci. Powyżej przedstawiono fragment programu w języku ASSEMBLER 8080, realizujący wymienione czynności oraz czas jego wykonania.

Najdłuższy okres oczekiwania na dane występuje w przypadku wykonania następującego ciągu instrukcji: CMP (5T), JNZ (10T), CMP (5T), JNZ (10T), IN (10T), CMP (5T), JNZ (10T) oraz IN (10T) — w sumie 65 cykli zegarowych T. Przy największej częstotliwości zegara, z jaką może pracować mikroprocesor 8080 (T=0,48 μs), daje to czas 31,2 μs. Najdłuższy czas realizacji zapisu jest jeszcze większy — ze względu na konieczność pobrania danych z pamięci przed przesłaniem ich do układu.

Przedstawiony przykład pokazuje jak krytyczny jest czas, w którym mikroprocesor musi obsłużyć układ 8271 podczas transmisji danych. Poniżej przedstawiony zostanie zarys programu rozwiązującego problem (druga część wydruku).

Program wymaga odpowiedniego ustawienia poszczególnych fragmentów w pamięci, przy czym niektóre jego części muszą być powtórzone. W tym przypadku za czas wykonania trzeba zapłacić obszarem zajmowanym w pamięci.

Przy tak napisanym programie udało się uzyskać dla najgorszego przypadku przy odczycie danych czas odpowiadający 50 cyklom zegarowym, a przy zapisie — 57 cyklom zegarowym.

Program obsługi dysków elastycznych powinien być dostosowany do potrzeb przewidywanego systemu operacyjnego. Systemy operacyjne z reguły dokonują transmisji pomiędzy pamięcią a dyskiem, poprzez własny bufor danych o długości jednego sektora. Pojedyncza operacja dyskowa dotyczy z reguły tylko jednego sektora, a po jej zakończeniu przygotowywana jest kolejna zawartość bufora. Takie działanie powoduje znaczne wydłużenie czasu współpracy z dyskami, gdyż podczas przygotowywania bufora dyskietka zdąży obrócić się na tyle, że kolejny sektor minie już głowicę odczytującą. Dostęp do tego sektora możliwy będzie dopiero po pełnym obrocie dysku (ok. 1/6 s). Przyspieszenie operacji dyskowych można uzyskać przez fizyczną lub logiczną zmianę numeracji sektorów, czyli przez sformatowanie ścieżki w taki sposób, by sektory nie występowały w kolejności swoich numerów (ale np. co drugi lub co trzeci) lub przez nie zapisywanie zbioru w kolejnych sektorach.

Znacznie lepsze wyniki można byłoby uzyskać dokonując jednoczesnej operacji na kilku sektorach, na co pozwala układ 8271, lub przez zrównoleglenie przesyłania danych kanałem DMA i przygotowywanie drugiego bufora danych — tak, by kolejną operację można było rozpocząć natychmiast po zakończeniu poprzedniej. Wybór rozwiązania zależy jednak od specyfiki konkretnego systemu operacyjnego.

ADAM PLUTA
Warszawa

Nazbyt może skrótowy opis systemu DS-80 MONITOR zamieszczamy z nadzieją, że mimo wszystko pozwoli się on zorientować w podstawowych możliwościach systemu. Koncepcja zastosowana w DS-80 znacznie odbiega od przyjętej w popularnych systemach operacyjnych; jest ona — jak się wydaje — szczególnie atrakcyjna dla mikrokomputerów nie posiadających szybkich pamięci masowych, czyli — korzystających z magnetofonu.

Jeszcze raz powtarzamy zasadę: nie marnujmy czasu na robienie tego, co już zostało zrobione! Opisane oprogramowanie praktycznie każdy może uzyskać na dogodnych zasadach (nawet w wersji źródłowej). Oczywiście, odbiorcy profesjonalni muszą liczyć się z pewnymi nakładami. Pod tekstem podajemy numer telefonu autorów.

DS-80 MONITOR

Jako podstawowe urządzenia peryferyjne systemu, w którym można wykorzystać program DS-80 MONITOR stosowane są: monitor z wejściem wizyjnym, klawiatura alfanumeryczna, magnetofon kasetowy.

Program mieści się w pamięci EPROM o pojemności 4 KB i korzysta z 4 KB pamięci RAM przeznaczonych na pamięć obrazu i obszar roboczy.

DS-80 MONITOR umożliwia realizowanie następujących operacji:

- odczytywanie zdisasemblowanej zawartości pamięci w standardowej formie edycyjnej (adres — kod maszynowy — etykieta — mnemonik)
- wprowadzanie programów w języku ASSEMBLER z możliwością stosowania etykiet, adresowania bezwzględnego, względnego i symbolicznego
- wpisywanie do pamięci liczb w kodzie szesnastkowym
- relokowanie, rozsuwanie (w celu dopisania dodatkowych rozkazów) lub usuwanie zbędnych fragmentów programu z jednoczesną korekcją adresów
- uruchamianie programów użytkowych od dowolnego adresu
- realizację pracy krokowej (z możliwością modyfikacji zawartości rejestrów procesora); praca krokowa może być prowadzona automatycznie aż do żądanego adresu lub zadanej liczby przejść przez taki adres
- wprowadzanie programów użytkowych z magnetofonu kasetowego i zapis zawartości pamięci na kasetę (modulacja amplitudy; 600 bodów).

Odczytywanie zawartości pamięci rozpoczyna się od zadanego adresu. Adres może być deklarowany w postaci absolutnej (kod szesnastkowy) lub symbolicznej (etykieta), o ile została ona wcześniej zadeklarowana (i co za tym idzie — wprowadzona do zbioru etykiet). Zawartość pamięci traktowana jest jako program wynikowy, który podlega disasemblowaniu. Przy disasemblowaniu uwzględniana jest tablica etykiet i w przypadku zgodności adresów — etykiety wprowadzane są w pole etykiety lub jako operandy instrukcji 3-bajtowych. Zawartość pamięci wyświetlana jest na ekranie monitora w wierszach po jednej instrukcji. Kolejne wiersze wyświetlane są u dołu ekranu po przepisaniu w górę poprzedzających wierszy. Na ekranie mieści się 26 wierszy zawierających nie więcej niż 64 znaki. Wyprowadzanie odbywa się tak długo, jak długo wciśnięty jest klawisz SPACE. Może być ono wznowione po ponownym wciśnięciu tego klawisza. Zadeklarowanie nowego adresu powoduje wymazanie poprzedniej zawartości ekranu.

Wprowadzanie programów lub danych może być realizowane w postaci symbolicznej lub w kodzie szesnastkowym. Przed rozpoczęciem wpisywania należy zadeklarować adres początkowy. Rozkazy w postaci symbolicznej wpisuje się zgodnie z normami języka ASSEMBLER. Argumenty rozkazów 3-bajtowych mogą być wprowadzone w kodzie szesnastkowym, w postaci symbolicznej (etykiety) lub względnej w zakresie od — 99 do 99 bajtów. Etykiety muszą być wcześniej umieszczone w zbiorze etykiet. Po skompletowaniu rozkazu, w wyniku asemblacji — w odpowiednie komórki pamięci zostaje wpisany wynikowy kod maszynowy. W przypadku gdy nowo wpisany kod maszynowy nie pokrywa wszystkich bajtów rozkazu dotychczas występującego w tym miejscu, pozostałe bajty zostają zastąpione rozkazami NOP.

Wpisanie nowej etykiety w miejsce już istniejącej zmienia etykietę przyporządkowaną adresowi wpisywania. Etykieta może być usunięta przez wpisanie samego dwukropka. Maksymalna liczba etykiet w podstawowej wersji programu wynosi 145. Tworzą one zbiór zajmujący 1 KB pamięci.

Zamiast rozkazów w postaci symbolicznej, można wprowadzać liczby szesnastkowe złożone z dwóch lub czterech znaków zakończonych przecinkiem. W przypadku wpisania czterech znaków zmieniana jest kolejność bajtów, tak jak przy pseudorozkazie DW.

Rozkazy w postaci symbolicznej mogą być wpisywane tylko w przypadku, gdy adres wpisywania odpowiada pierwszemu bajtowi rozkazu dotychczas znajdującego się w pamięci. Liczby szesnastkowe mogą być wpisywane w dowolne miejsce.

W procesie wpisywania kontrolowana jest formalna poprawność zapisu; w przypadku wystąpienia jakichkolwiek błędów — są one sygnalizowane napisem ERROR, a błędne zlecenie nie jest wykonywane.

Zastosowany sposób wprowadzania daje umieszczenie w pamięci mikrokomputera absolutnego programu wynikowego, który może być uruchomiony natychmiast po zakończeniu wpisywania.

Programy mogą być przechowywane na kasetach magnetofonowych. Przed wyprowadzeniem programu należy zadeklarować jego adres początkowy i końcowy. Razem z programem na taśmę zapisywana jest także zawartość zbioru etykiet.

Uzupełnieniem programu operacyjnego są wprowadzane z taśmy magnetofonowej procedury:

— drukowania zawartości pamięci w standardowej formie edycyjnej za pomocą drukarki DZM 180 KSR.

— drukowania zawartości pamięci z możliwością umieszczenia komentarza — programowania pamięci EPROM typu 2708 i 2716.

Opracowano także rozszerzoną wersję programu o objętości 5 KB, w której przewidziano możliwość specjalnego wyprowadzania tabel oznaczonych etykietami rozpoczynającymi się od liter TB (tablica danych 1-bajtowych), TW (tablica danych 2-bajtowych) i TA (tablica tekstów w kodzie ASCII). Zwiększono także liczbę etykiet do 255, dołączono program drukowania oraz wprowadzono możliwość sprawdzania poprawności zapisu programu na taśmie magnetofonowej. Podobną wersję programu operacyjnego opracowano także dla mikroprocesora Z-80 (Z-80.S MONITOR). Ze względu na bardziej złożony ASSEMBLER, program ten ma objętość 6 KB.

Jak wynika z doświadczeń, omówione programy operacyjne są bardzo dobrym narzędziem programowania w języku ASSEMBLER oraz uruchamiania i testowania programów użytkowych. Ich modularna struktura umożliwia wykorzystywanie zawartych w nich podprogramów przez programy użytkowe. Dotyczy to obsługi operacji we-wy, konwersji kodów, pętli opóźniających, obsługi transmisji DMA i innych tego typu operacji. Program jest szybki: przesuwanie bloku o objętości 4 KB przy maksymalnej liczbie etykiet (255) wraz ze zmianą adresów trwa nie dłużej niż kilka sekund. Praca z systemem jest znacznie szybsza i wygodniejsza niż przy użyciu firmowego EDITORA i ASSEMBLERA.

Mankamentem systemu jest brak możliwości wprowadzania komentarzy podczas pisania programu; można je wprowadzać jedynie w trakcie końcowej dokumentacji programu.

GRZEGORZ STĘPIEŃ
MIROSŁAW DOLEŻYCH
JERZY ORKISZEWSKI
Politechnika Warszawska, IBSPiE
tel. 49-98-71 w. 538

Konsekwentnie wprowadzana reforma spowodowała, że wiele rzeczy (często podstawowych) ma zbyt wysoką cenę jak na możliwości przeciętnego obywatela. O mikrokomputerach trudno jednak powiedzieć, że są drogie. Są po prostu poza możliwościami finansowymi znakomitej części społeczeństwa.

Propozycja kupowania używanych komputerów oczywiście nie rozwiązuje tego problemu. Staramy się jednak nie zaprzepaścić żadnej, nawet najdrobniejszej, szansy upowszechniania mikroinformatyki.

Komputery z drugiej ręki

W poszukiwaniu taniego (a więc odpowiadającego naszym zasobom gotówki) mikrokomputera bezsprzecznie należy rozważyć możliwość zakupu używanego egzemplarza. Należy jednak przedtem znaleźć odpowiedź na kilka podstawowych pytań:

— skąd pochodzi używany komputer

— czy będzie pracował jako całość czy też należy go raczej potraktować jako źródło części zamiennych

— czy dostępne są instrukcje obsługi, czy też wszystko trzeba będzie wynajdywać metodą Sherlocka Holmesa

— czy jego wartość odpowiada cenie.

Gwałtowny rozwój mikroinformatyki znajduje swoje odbicie w stosunkowo częstej wymianie sprzętu przez użytkowników. Szczególnie przez tych profesjonalnych. Dla nadążenia za postępem w tej dziedzinie zalecana jest wymiana komputera mniej więcej co dwa lata (!). Wiele instytucji próbuje jednak pogodzić to zalecenie z zasadami ekonomiki czyli wycisnąć z komputera ile tylko się da — dopóki jego eksploatacja jest opłacalna (szczególnie te mniej zamożne — na przykład uczelnie). Z kolei niektóre komputery spełniają głównie rolę eksponatów i można powiedzieć, że są praktycznie nowe. Tak więc w zależności od rodowodu używany komputer może reprezentować różną wartość.

Tańsze mikrokomputery, przeznaczone w zasadzie do domowego użytku przez prywatnych odbiorców (na przykład: VC 20, ORIC, ZX SPECTRUM, C 64, ATARI), różnią się między sobą niezawodnością konstrukcji lecz generalnie rzecz biorąc nie są one produkowane z myślą o eksploatacji przez długie lata. Co gorsza, wiele modeli jest wprowadzanych na rynek z dużym pośpiechem, bez dokładnego przetestowania konstrukcji i dopiero druga albo nawet trzecia wersja jest wolna od błędów. Czasem szwankuje współpraca z urządzeniami peryferyjnymi, innym razem okazuje się, że nie dopracowano oprogramowania firmowego, a kłopoty z przegrzewaniem się mikrokomputerów należą prawie do reguły (np. ZX SPECTRUM, APPLE II).

Jeżeli dostępny jest komputer z poprawionej wersji to można oczekiwać, że przez parę ładnych lat będzie funkcjonował praktycznie niezawodnie. Mogą jednak wystąpić trudności z rozpoznaniem wersji z którą mamy do czynienia. Tylko sporadycznie, jak na przykład w przypadku komputera OSBORNE, producent wprowadza jakiś element rozróżniający (tu zmieniono kolor obudowy). Przykładowo, w ZX SPECTRUM trzeba by zdjąć obudowę aby przekonać się, że na płytce drukowanej umieszczono napis „issue 3".

Z dobrym przybliżeniem można jednak założyć, że nawet w przypadku trafienia na nie dopracowaną wersję wszystkie układy scalone będą sprawne i jeżeli znamy istotę wprowadzonych przez producenta poprawek, to często możemy dokonać zmian we własnym zakresie.

Po kilkuletniej niezawodnej eksploatacji wzrasta jednak prawdopodobieństwo pojawienia się usterek. Po pierwsze mogą wystąpić mikropęknięcia wokół wyprowadzeń układów scalonych, umożliwiające wniknięcie pary wodnej i rozpoczęcie procesów korozji. Zdarzają się też pęknięcia ścieżek drukowanych lub też uszkodzenia układów scalonych, wywołane przez zwężenie się wewnętrznych mikropołączeń w wyniku migracji nośników elektrycznych.

W przypadku sprzętu profesjonalnego szacuje się, że po ok. siedmiu latach usterki mogą nasilić się do tego stopnia, że dalsza eksploatacja przestaje być opłacalna.

Najbardziej podatnymi na zużycie częściami mikrokomputera są wszelkie elementy elektromechaniczne, a więc klawiatura, napędy dysków elastycznych, drukarki lub magnetofony. W przypadku monitorów ekranowych dość częstym zjawiskiem jest wypalenie lampy kineskopowej.

Jeżeli podejmiemy decyzję o zakupie używanego komputera to istnieją dwie możliwości:

— zakupić odpowiednie pismo fachowe lub hobbystyczne i wśród prywatnych ogłoszeń znaleźć interesującą nas ofertę

— wybrać się do sklepu prowadzącego skup i sprzedaż używanego sprzętu komputerowego.

W pierwszym przypadku występuje dosyć duże ryzyko związane z tak zwanymi wadami ukrytymi. Niektóre uszkodzenia dają o sobie znać dopiero po pewnym czasie, inne tylko w specyficznych sytuacjach. Trzeba więc mieć trochę doświadczenia i przysłowiowy łut szczęścia.

W przypadku sklepu występuje trochę inna sytuacja: przed odsprzedażą sprzęt poddawany jest testom, gdyż zazwyczaj udzielana jest miesięczna gwarancja. Nie eliminuje to oczywiście możliwości wystąpienia uszkodzeń, ale jednak poważnie ją zmniejsza. Oczywiście za pośrednictwo i testowanie pobierana jest opłata i wynosi ona około 30%. Do tego Brytyjczycy muszą jeszcze doliczyć 15% podatku. Przybysze z Polski po otrzymaniu na granicy potwierdzenia wywozu, mogą ze stratą w wielkości opłat manipulacyjnych odzyskać zapłaconą kwotę podatku (tzw. VAT — ang. Value Added Tax).

A oto przykładowe ceny (w funtach) używanych mikrokomputerów (łącznie z podatkiem) — listopad 1984:

BBC model B	— 325
ACORN ATOM	— 40
COMMODORE C64	— 139
EPSON HX-20	— 325
DRAGON	— 65
NEWBRAIN A	— 103
SPECTRUM	— 99
JUPITER ACE	— 15
LYNX	— 115
NEWBRAIN AD	— 110
PC-1500	— 65
z drukarką i magnetofonem	75

Warto zwrócić uwagę, że na przykład całkiem niezły mikrokomputer ACORN ATOM po prostu wyszedł z mody i w porównaniu do innych oferowany jest prawie za bezcen. Wadą JUPITERA ACE jest fakt, że zamiast interpretera języka BASIC ma on zaimplementowany FORTH. Jednak ci, co potrafią programować w tym języku na pewno będą odmiennego zdania i mimo niewielkiej pamięci RAM (16 KB) zdecydują się na zapłacenie tej prawie symbolicznej ceny.

Oprócz komputerów, w sklepach z używanym sprzętem często można nabyć naprawdę okazyjnie urządzenia peryferyjne. Oto przykłady. Ośmiocalowy dysk Winchester o pojemności 19 MB — praktycznie nowy — oferowany był przez jeden z londyńskich sklepów za... 280 funtów. Drukarka firmy IBM GOLFBALL sprzedana była za 80 funtów.

Niektóre urządzenia niezłej klasy sprzedawane są po niskiej cenie z absurdalnych, jak na nasze wyobrażenia, powodów. Na przykład, ploter bębnowy CALCOMP o dokładności 1/100 cala — i to nie tylko na szerokości papieru, ale i wzdłuż całej rolki o długości 36 m — można było kupić za 165 funtów. Jednak linie interfejsu tego plotera zamiast standardowych poziomów TTL wymagają sterowania napięciem rzędu 10V.

Można oczekiwać, że podane informacje wzbudzą wiele emocji wśród Czytelników, wybierających się właśnie na Zachód. Niestety redakcja dysponuje tylko telefonami sklepów z używanym sprzętem komputerowym na terenie Wielkiej Brytanii (a i to zapewne nie wszystkich). Osoby, które mają podobne informacje z innych krajów prosimy o udostępnienie swoich doświadczeń (za naszym pośrednictwem) pozostałym Czytelnikom.

AJP

W Akademii mikroKLANU nie staramy się uczyć programowania w języku BASIC. Znaczenie poszczególnych instrukcji tego języka można znaleźć w dołączanej do każdego mikrokomputera instrukcji obsługi. Na dodatek istnieje niemal tyle odmian tego języka ile typów mikrokomputerów.

Akademię mikroKLANU adresujemy do tych, którzy mają już za sobą pierwsze poprawnie działające programy. Chcemy im pokazać na czym polega elegancja w pisaniu programów. Bardziej doświadczonych Czytelników zapraszamy do współpracy w Akademii.

Akademia mikroKLANU (3)

Koniunkcja

W tym odcinku zajmiemy się zasadniczą częścią programu czyli tą, w której wykonywane są manipulacje na danych. Pamiętamy już, że program powinien być nie tylko łatwy w obsłudze lecz także zrozumiały dla innych — jest to tzw. **reguła kumpla**. Poniższy program, aczkolwiek może zostać zrealizowany przez większość komputerów nie wygląda zbyt zachęcająco:

```
LIST

10 INPUT"DODATNIA LICZBA CALKOWITA=";N
20 M=N
30 IFN=2THENGOTO80
40 N=N/2
50 IFINT(N)=NTHENGOTO30
60 N=N*6+1
70 GOTO30
80 PRINT"LICZBA ";M;" SPELNIA ZASADE ULAMA!"
90 END
READY.
```

A przecież opisane w poprzednich odcinkach reguły tworzenia wejścia i wyjścia były tu skrupulatnie przestrzegane! Autor tego programu posunął się za daleko w oszczędności pamięci komputera i przez to zupełnie zatracił czytelność programu. Niestety wiele osób uważa, że pisanie oszczędnych programów polega tylko na oszczędzaniu pamięci komputera. Szkoda, że nie potrafią oszczędzać własnego i cudzego czasu... W przykładzie brak też jakichkolwiek komentarzy co program robi. Zastanówmy się, jakie należy wprowadzić zmiany, by program stał się w miarę czytelny.

Po pierwsze należy pisać z odstępem między instrukcjami — (niektóre interpretery języka BASIC automatycznie wstawiają spacje między instrukcje).

Po drugie komentarze powinny towarzyszyć większości operacji (oczywiście nie trzeba komentować faktu dzielenia przez dwa lecz raczej podać **kiedy** program dzieli przez dwa, jeżeli nie jest to oczywiste).

Wygodnie jest umieszczać komentarze w linijkach z numerami nie będącymi wielokrotnością 10 (za to instrukcje tylko w liniach o numerach będących wielokrotnością 10). W ten sposób ktoś, kto będzie przepisywał nasz program i ewentualnie przystosowywał go dla innego komputera, może łatwo pominąć niektóre komentarze uzyskując dodatkowe miejsce dla dokonania poprawek wynikających ze specyfiki konkretnego komputera.

Dobrze jest też zamieszczać w programie instrukcje wyjścia w środku pętli. W przykładowym programie już dla N równego 31, czas wykonania programu jest dość znaczny — warto więc sygnalizować użytkownikowi, że program liczy. Przedłuża to nieco czas realizacji programu, ale daje pewność, że program działa poprawnie. Jest to szczególnie istotne przy testowaniu poprawnej pracy programu. Przykładowy program można więc uzupełnić:

```
75 PRINT N;
95 PRINT N;
```

Gdy przekonamy się, że program działa poprawnie, można linie 75 i 95 opuścić.

Warto umieścić w programie kilka instrukcji zbierających dane statystyczne (np. o liczbie cykli, które zostały wykonane i największej liczbie, do której doszliśmy w naszej procedurze.

Warto też, po dokonaniu wszystkich poprawek, przenumerować linie programu — znowu zyskamy instrukcje tylko w linijkach mających numery będące wielokrotnością dziesięciu.

Tak więc w rezultacie przykładowy program ma następującą postać:

```
LIST

5 REM PROGRAM SPRAWDZA ZASADE ULAMA:JEZELI DODATNIA
6 REM LICZBA CALKOWITA JEST PARZYSTA TO NALEZY PODZIELIC
7 REM JA PRZEZ 2,GDY JEST NIEPARZYSTA TO POMNOZYC PRZEZ 3
8 REM I DODAC 1 A CALA PROCEDURA POWTARZYC AZ DO OTRZYMANIA
9 REM LICZBY 1 W SKONCZONEJ ILOSCI KROKOW.
10 INPUT "DODATNIA LICZBA CALKOWITA=";N
20 IF N<1 OR INT(N)<>N THEN GOTO 10
30 M=N
40 I=0
41 REM I JEST LICZNIKIEM ILOSCI CYKLI
50 X=1
51 REM X JEST NAJWIEKSZA OSIAGNIETA LICZBA
60 I=I+1
70 IF N=2 THEN GOTO 130
71 REM JEZELI N=2 TO  NASTEPNYM KROKU BYLOBY ROWNE 1
80 N=N/2
90 IF INT(N)=N THEN GOTO 60
91 REM POPRZEDNIE N BYLO PARZYSTE
100 N=N*6+1
101 REM POPRZEDNIE N BYLO NIEPARZYSTE I ZOSTALO JUZ PODZIELONE PRZEZ 2
110 IF N>X THEN X=N
120 GOTO 60
130 PRINT "LICZBA ";M;" SPELNIA ZASADE ULAMA!"
140 PRINT "ILOSC CYKLI DO OSIAGNIECIA 1 WYNOSI ";I
150 PRINT "NAJWIEKSZA OSIAGNIETA LICZBA WYNOSI ";X
160 END
READY.


RUN
DODATNIA LICZBA CALKOWITA=? 31
LICZBA  31  SPELNIA ZASADE ULAMA!
ILOSC CYKLI DO OSIAGNIECIA 1 WYNOSI  106
NAJWIEKSZA OSIAGNIETA LICZBA WYNOSI  9232

READY.
```

W tym programie niestety nie udało się nam uniknąć stosowania instrukcji bezwarunkowej **GOTO** (linia 120). Teoretycy informatyki (patrz np. N. Wirth „Wstęp do programowania systematycznego". WNT, 1978) uważają, że dobry program nie powinien zawierać takich instrukcji. W języku BASIC podprogramy są też wywoływane przez numer linii, a więc strukturalnizacja programu niewiele polepsza jego zrozumiałość. Ale o tym będzie mowa w następnych odcinkach!

J. TATARKIEWICZ

Tak zwana zasada Ulama wcale nie została podana przez tego wybitnego matematyka polskiego pochodzenia! On sam mówi o niej: „Jest ona taka piękna, że chciałbym być tym co ją wymyślił!" Z dostępnej literatury wynika, że zasada ta występuje pod różnymi nazwami (algorytm z Syrakuz, problem Collatza-Kakutani i in.). Do dzisiaj nie tylko nie podano dowodu, iż zasada jest poprawna dla wszystkich liczb całkowitych, ale także nie ma dowodu, że koniunkcji tej nie da się udowodnić. Dr Bryan Thwaites z Westfield College w Anglii ufundował nagrodę 1000 funtów dla każdego kto przedstawi poprawny dowód zasady lub uzasadnienie niemożliwości takiego dowodu. Wiadomo, że przetestowano już liczy aż do około 2^{50} i zawsze zasada była prawdziwa. A może ktoś z Czytelników INFORMATYKI zdobędzie nagrodę?

Ceny sprzętu komputerowego w Anglii, marzec 1985 (w funtach — bez podatku V.A.T.)

COMMODORE C64 — 166.—; COMMODORE C16 (+ magnetofon) — 100.—; COMMODORE PLUS 4 — 200.—; COMMODORE 1541 (stacja dysków) — 160.—; COMMODORE 801 (drukarka) — 130.—; COMMODORE 802 (drukarka) — 261.—; ACORN ELECTRON — 110.—; BBC (model B) — 290.—; SINCLAIR QL (128KB) — 335.—; SPECTRUM 48K PLUS 107.—; MICRODRIVE + INTERFACE1 — 83.—.
Obudowa zamieniająca SPECTRUM w SPECTRUM PLUS — 20.—
AMSTRAD CPC 464 (+ monitor monochromatyczny) — 209.—
AMSTRAD CPC 464 (+ monitor kolorowy) — 301.—
SERIA MSX:
TOSHIBA HX10 64K — 190.—; SANYO MCR10 64K — 235.—; SONY HB75B 64K —245.—; JVC HC7GB 64K — 226.—.
DRUKARKI:
SEIKOSHA GP500A — 113.—; SEIKOSHA GP700A Colour — 244.—; SHINWA CTA80 — 200.—; EPSON RX80.

Ceny podzespołów w RFN, luty 1985 (w markach z podatkiem MWST)

PAMIĘCI:
2114 — 7.95, 4116 (200) — 3.95, 4164 (250) — 13.50, 41256 (150) — 79.—, 2708 — 8.70, 2716 (450) — 13.80, 2732 (350) — 18.90, 2732 (450) — 15.90, 2764 (250) — 21.—, 27256 (250) — 195.—,
MIKROPROCESORY:
Z80A — 6.50, Z80B — 15.—, 6502P — 13.90, 6502A — 15.25, 6800 — 10.30, 6802 — 9.95, 6809 — 22.90, 68000 (8MHz) — 82.—,
STEROWNIKI CRT:
6845 — 16.50, EF 9365(96)7 — 85.—, UPD 7220 — 79.—
STEROWNIKI Nnapędów DYSKOWYCH:
1771 — 39.—, 1791 — 27.50, 1797 — 27.50, 2793 — 65.—, 2797 — 65.—, 1691 — 26.50, 9229B — 89.—

- W Tarnowskich Górach od października 1984 rozpoczął działalność Klub Mikrokomputerowy, który zadomowił się w lokalu Klubu Technika w dzielnicy Strzybnica. Kontakt: Piotr Drewniak (prezes), ul. W. Wasilewskiej 4/5; 42-609 Tarnowskie Góry tel. d. 85-83-01, tel. praca 85-22-41 w. 585, (kierunkowy jak na Katowice)
- Przy ZR Kasprzaka w Warszawie rozpoczął działalność klub mikrokomputerowy „Mikros". Informacje: Waldemar Kufel, tel. p. 32-99-27 lub 32-16-21
- Zapowiadane broszury mikroKLANU (kilka jest już gotowych!) zapewne ukażą się... nie wcześniej jak w roku 1986. Naiwnie sądziliśmy, że broszura drukuje się najwyżej kilka miesięcy.

Przedsiębiorstwo Polonijno-Zagraniczne ITM

31-546 Kraków, ul. Mogilska 80

poleca

sprzęt komputerowy z oprogramowaniem

- komputer osobisty MERITUM I (16, 32 lub 48K RAM)
- drukarka Seikosha typ GP-500A
- generator w.cz. umożliwiający podłączenie Meritum do dowolnego odbiornika TV
- biblioteka programów matematycznych i statystycznych
- oprogramowanie Asembler/Edytor dla Z80 oraz INTEL 8035
- oprogramowanie specjalne na zamówienie użytkownika

EO/391/K/85

prowadzi
ANDRZEJ J. PIOTROWSKI
tel. dom.: 48-22-85